Amendments to the Specification

*Please **replace** paragraph [0001] with the following amended paragraph:*

BACKGROUND OF THE INVENTION

[0001] Redundant array of inexpensive (or independent) disks (RAID) is an evolving data storage technology that offers significant advantages in performance, capacity, reliability, and scalability to businesses that have demanding data storage and access requirements. In 1988, a paper was published by Patterson, Gibson, Katz, entitled "A Case for Redundant Arrays of Inexpensive Disks (RAID)," International Conference on Management of Data, pages 109-116, June 1988. This paper described how RAID data storage would improve the data input/output (I/O) rate over that of a comparable single disk data storage system, ~~but~~ **and** how RAID data storage would provide fault tolerance, i.e., the ability to reconstruct data stored on a failed disk.

*Please **replace** paragraph [0002] with the following amended paragraph:*

[0002] RAID data storage systems are configured according to any one of a number of "RAID levels." The RAID ~~levels~~ **level** specify how data is distributed across disk drives in the array. In the paper noted above, the authors describe RAID ~~level~~ **levels** 1 through ~~level~~ 5. Since the publication of the paper mentioned above, additional RAID levels have been developed.

*Please **replace** paragraph [0016] with the following amended paragraph:*

DETAILED DESCRIPTION

[0016] The present invention provides for application assisted recovery from corruption of data using successive rereads. The present invention will be described with reference to correcting corrupted data in a volume stored in a RAID level 5 data storage system, it being understood that the present invention should not be limited thereto. Figure 1 shows relevant components of an exemplary RAID level 5 data storage system 10 employing one embodiment of the present invention. Data storage system 10 is coupled to a computer system 12 and configured to receive

requests to read data therefrom. The term coupled should not be limited to what is shown in Figure 1. Two devices, e.g., data storage system 10 and computer system 12, may be coupled together via a third device (not shown). Although not shown, data storage system 10 may be coupled **to** other computer systems and configured to receive requests to read data therefrom. Copending U.S. Application No. **10/195,679** ~~(Attorney Docket No. VRT0061US)~~ entitled "Automated Recovery From Data Corruption Of Data Volumes In RAID Storage" filed July 1, 2003, describes additional data storage systems and is incorporated herein by reference in its entirety.

*Please **replace** paragraph [0017] with the following amended paragraph:*

[0017] Data storage system 10 includes disk drives 16(1)-16(5), each of which is individually coupled to a RAID controller 18. The present invention can be implemented with a system that includes more or fewer disk drives than that shown in Figure 1. RAID controller 18 is capable of simultaneously reading data from one or a number of disk drives 16(1)-16(5) in response to a request received from computer system 12. RAID controller 18 may take form in a computer system having one or more processors that process data in accordance with instructions stored in a computer readable medium **The present invention as described herein, can be implemented by RAID controller 18 executing instructions stored in a computer readable medium such as memory 19**.

*Please **replace** paragraph [0025] with the following amended paragraph:*

[0025] If RAID controller 18 determines in step 34 that table 32 includes an entry matching $B_{y,x}$, then the process proceeds to step 36 where RAID controller compares the current time T_c with time stamp $T_{y,x}$ of the entry containing $B_{y,x}$. If current time T_c is more than a predetermined amount of time T_D greater than $T_{y,x}$, the entry is deleted from table 32 as being stale, and the process proceeds to step 46. If, in step 36 the entry is not stale (i.e., current time T_c is not more

than a predetermined amount of time T_D greater than $T_{y,x}$), ~~**new**~~ reconstructed data is generated for stripe unit $B_{y,x}$ using parity P_y and data of stripe S_y other than stripe unit $B_{y,x}$, as shown in step 42. The ~~**newly generated**~~ **reconstructed** data of stripe unit $B_{y,x}$ is returned to the computer system in step 44. **Optionally, the reconstructed data is compared with the existing data of $B_{y,x}$, and a notification is sent to computer system 12 if the reconstructed data does not compare equally**. Optionally, old data of stripe unit $B_{y,x}$ is overwritten with the ~~**newly generated**~~ **reconstructed** data.

Please ***replace*** *paragraph [0031] with the following amended paragraph:*

[0031] If table 62 does include a matching entry (i.e., an entry containing stripe unit quantity m and first stripe unit identity $B_{y,x}$**, where x takes on the value of the first unit of the set of m units**) in step 66, the process proceeds to step 80 where RAID controller 18 compares the current time T_c with $T_{y,x}$ of the matching table entry. If current time T_c is more than a predetermined amount of time T_D after $T_{y,x}$, the matching entry is deleted from table 62 as being stale, and the process proceeds to step 74. If, in step 80, time T_c is less than a predetermined amount of time T_D after $T_{y,x}$, the matching entry is not stale, and the process proceeds to step 84.

Please ***replace*** *paragraph [0032] with the following amended paragraph:*

[0032] In step 84, RAID controller determines whether any bit of **the** bit map in the matching entry is set to logical zero. If at least one bit is set to logical zero, the process proceeds to step 84 where variable z is set to the position of a bit set to logical zero in the bit map of the matching entry. In step 86, ~~**new**~~ **reconstructed** data is generated for stripe unit $B_{y,z}$ as a function of parity data P_y and data of stripe units $B_{y,1}$-$B_{y,4}$ other than $B_{y,z}$. In step 90, the ~~**newly generated**~~ **reconstructed** data and data of stripe units $B_{y,x}$-$B_{y,x+m-1}$, other than the old data for stripe unit $B_{y,z}$, is returned to the computer system 12 as a reply to the request in step 60. Thereafter, in step 92, RAID controller sets the bit in position z of the bit map to logical 1. If this returned data is

deemed valid by computer system 12, no new requests for the same data is generated by computer system 12.

Please ***replace*** *paragraph [0034] with the following amended paragraph:*

[0034] Optionally, RAID controller 18 may save a copy of the **reconstructed** data ~~newly generated~~ in step 86. RAID controller may also save the current value of z set in step 84. The current value of z may be saved as an additional component of the entry in table 62 that matches the identity of the group of stripe units that contain the data requested in step 60. RAID controller 18, before or after the entry deletion step 82, may overwrite old data for stripe unit $B_{y,z}$ with the saved, ~~newly generated~~ **reconstructed** data, where z is defined by the value stored in the matching entry.